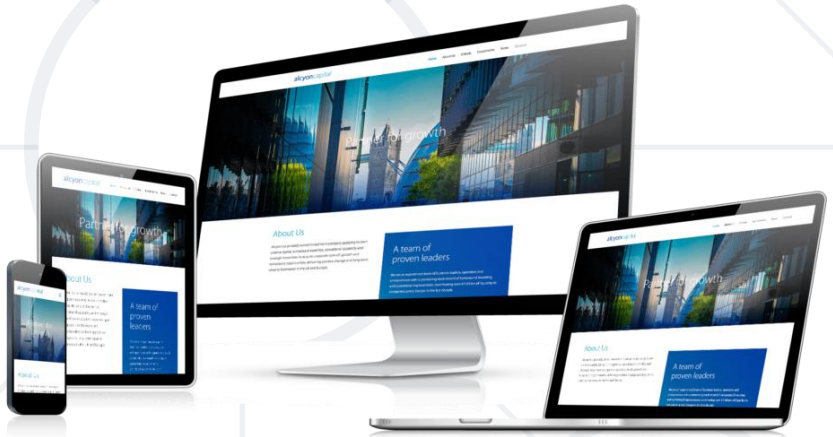
MEDIA QUERIES



SoftUni Team

Technical Trainers

Software University

SoftUni

Software University
https://softuni.bg

Table of Contents

Have a Question?

Software University

sli.do

#HTML-CSS

Responsive Web Design

What is Responsive Web Design?

- Responsive Web Design is about using HTML and CSS to **automatically resize**, hide, shrink, or enlarge, a website, to make it look good on **all devices** (desktops, tablets, and phones)
- Setting the Viewport – add the following <meta> element:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- This will give the browser instructions on how to control the page's **dimensions** and **scaling**

Components of Responsive Design

- Responsive website design consists of the following three main components:
  - **Flexible layouts** – Using a flexible grid to create the website layout. That will dynamically resize to any width
  - **Media queries** - Allow designers to specify different styles for specific browser and device circumstances
  - **Flexible Media** – Makes media (images, video and other format) scalable

Benefits of using a Responsive Website

- Increased **traffic** from mobile users
- Lower **cost** and website maintenance
- Provides a Seamless **User Experience**
- Adapts Easily To Any Screen Size
- A Responsive Website Improves Your **SEO** Efforts

Why Responsive Website Design Works?

Software University

- **Google** Prioritizes Responsive Websites
- 50% of Total **eCommerce** Revenue Comes from Mobile
- 94% of People **Judge** Websites on Responsive Web Design
- Almost 60% of All Internet **Access** is Done Through the Phone
- 77% of Adults Own a **Smart** Phone
- 72% of People Want **Mobile-Friendly** Websites
- Responsive Design Integrates **Social Media**

What is a Media Query?

- Media Queries are a feature of CSS that enable webpage content to **adapt** to different screen sizes and resolutions
- They are a fundamental part of **responsive web design** and are used to customize the appearance of websites for multiple devices

Media Queries

- Media queries in CSS3 look at the **capability** of the device
- Media queries can be used to check many things, such as:
  - width and height of the viewport
  - width and height of the device
  - orientation (is the tablet/phone in landscape or portrait mode?)
  - resolution

Media Query Syntax

- A media query consists of a **media type** and can contain one or more **expressions**, which resolve to either true or false

```
@media not|only mediatype and (media-feature-rule) {
  CSS rules goes here
}
```

- The result of the query is **true** if the specified media type matches the type of device the document is being displayed on
- Unless you use the **not** or **only** operators, the media type is **optional** and the all type will be implied

Media Query Syntax

- A **media type**, which tells the browser what kind of media this code is for (e.g. print, or screen)
- A **media feature rule** - test that must be passed for the contained CSS to be applied
- A set of **CSS rules** that will be applied if the test passes and the media type is correct


Desktop
Tablet
Smartphone
@media screen and
(min-width: 1024px)
{...}
@media screen and
(min-width: 768px) and
(max-width: 1023px)
{...}
@media screen and
(max-width: 767px)
{...}

Media Types

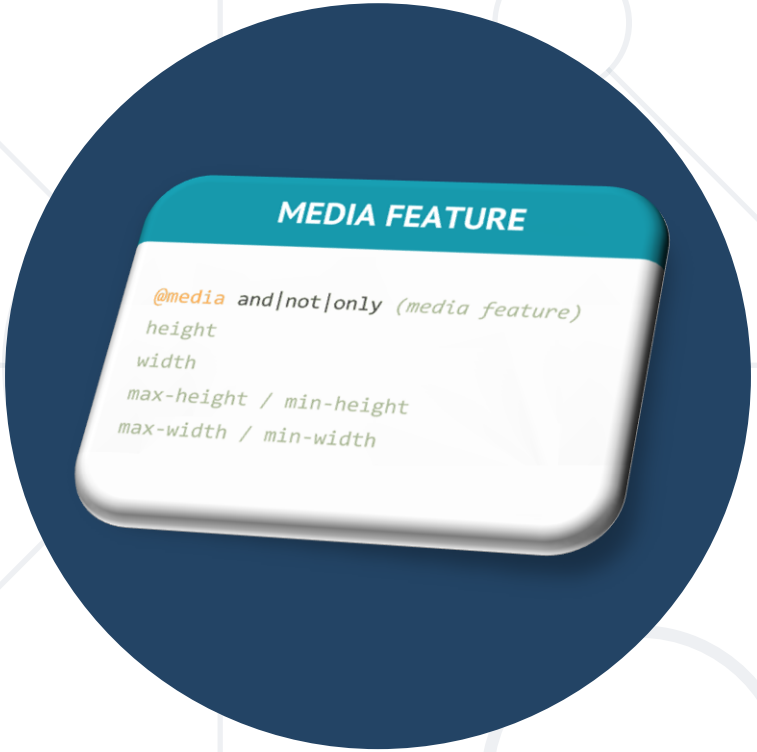
CSS Media Types

- Media Types describe the general category of a given device:
  - **All** – used for all media type devices
  - **Print** – used for printers
  - **Screen** – used for computer screens, tablets, smart-phones etc.
  - **Speech** – used for screenreaders that "reads" the page out loud

Media Query - Example

- The following media query will only set the body to 12px if the page is **printed**. It will not apply when the page is loaded in a browser:

```
@media print {
  body {
    font-size: 12px;
  }
}
```



Media Feature Rules

Media Feature Rules (1)

- After specifying the type, you can then target a media feature with a **rule**
  - Width and height – we can apply CSS if the viewport is above or below a certain width, using **width**

```
@media screen and (width: 600px) {
   body {
      color: red;
   }
}
```

Media Feature Rules (2)

- We can apply CSS if the viewport is with an exact width – using **min-width**, **max-width**

```
@media screen and (max-width:400px) {
  body {
    color: blue;
  }
}
```

Media Feature Rules (3)

- Orientation – allows to test for **portrait** or **landscape** mode
- To change the body text color if the device is in landscape orientation:

```
@media (orientation: landscape) {
  body {
    color: red;
  }
}
```

Media Feature Rules (4)

- Hover: used to query the user's ability to **hover** over elements on the page with the primary pointing device
  - hover: over elements with ease

```
@media (hover: hover) {
  /* ... *
}
```

  - none: can't hover over elements

```
@media (hover: none) {
  /* ... *
}
```

Media Feature Rules (5)

- Pointer: used to query the presence and accuracy of a **pointing** device such as a mouse
- If the primary input mechanism of the device includes:
  - a pointing device of limited accuracy, we use **coarse**

Media Feature Rules (6)

- an accurate pointing device, we use **fine**
- **not include** a pointing device, we use **none**

```
/* define styles based on input device pointer
accuracy */
@media(pointer: <course| fine | none>) {
    /* ... *
}
```

Media Feature Rules (7)

- Target specific existing device

```
/* smartphones, touchscreens */
@media (hover: none) and (pointer: coarse) {
/* ... */
}

/* stylus-based screens */
@media (hover: none) and (pointer: fine) {
/* ... */
}

/* mouse, touch pad */
@media (hover: hover) and (pointer: fine) {
/* ... */
}
```

Logical Operators

Logical Operators (1)

- The logical operators **not**, **and**, and **only** can be used to compose a complex media query
  - **AND** - combining multiple media features

```
@media screen and (min-width: 400px) and (orientation: landscape) {
    body {
        color: blue;
    }
}
```

Logical Operators (2)

- **NOT** - negate a media query

```
@media not all and (orientation: landscape) {
    body {
        color: blue;
    }
}
```

- **ONLY** - used to apply a style only if an entire query matches
- **,(comma)** - commas are used to combine multiple media queries into a single rule

@import

@import

- The **@import CSS at-rule** is used to import style rules from other style sheets
- You can specify media-dependent @import rules
- These conditional imports specify comma-separated media queries after the URL

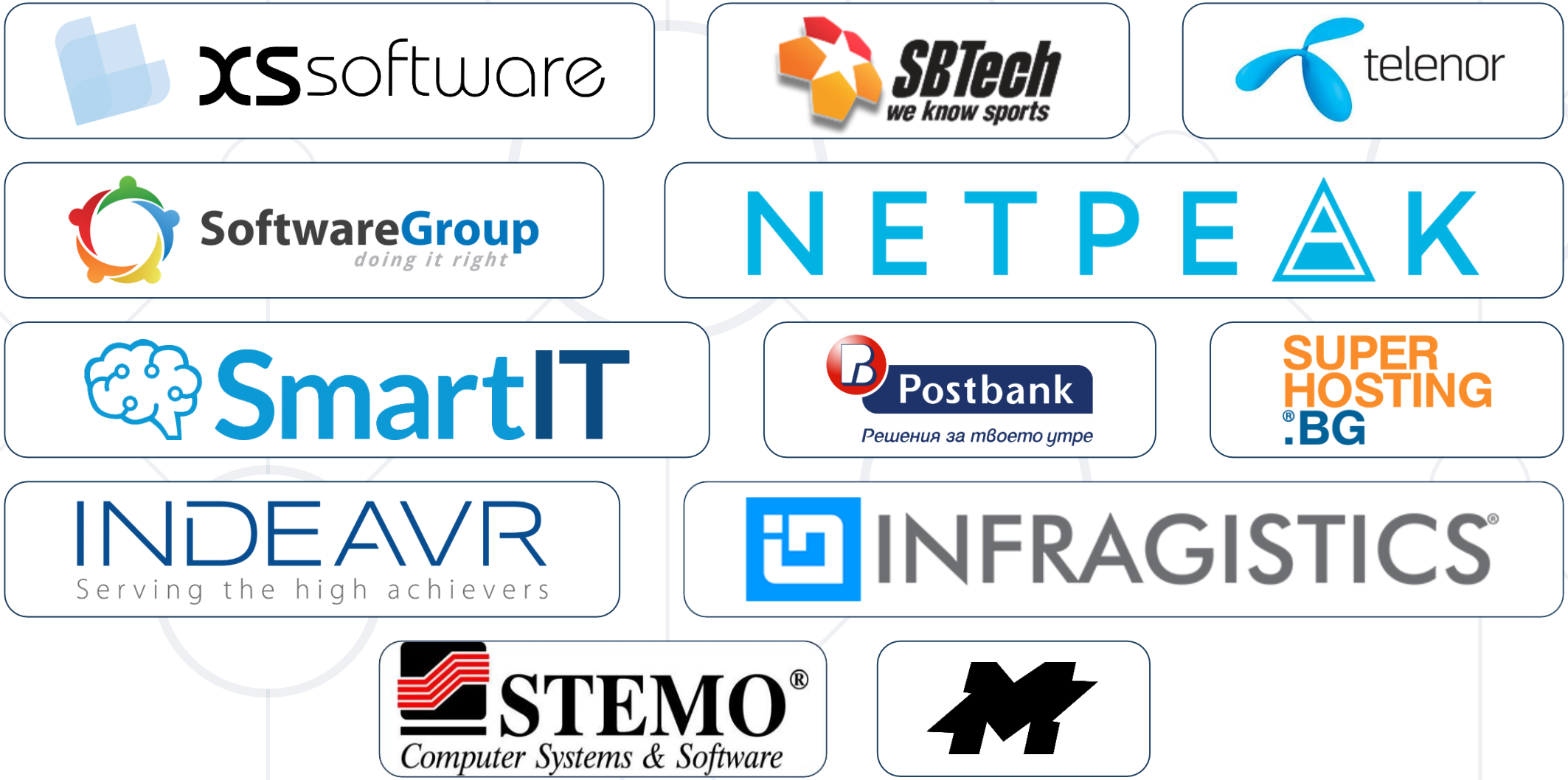
```
@import url;
@import url list-of-media-queries;
@import url("fineprint.css") print;
@import url(“landscape.css") screen and (orientation: landscape);
```

Summary

- What is Responsive Web Design?
- What are Media Queries?
- Media Types
- Media Feature Rules
- Logical Operators
- How to use @import?

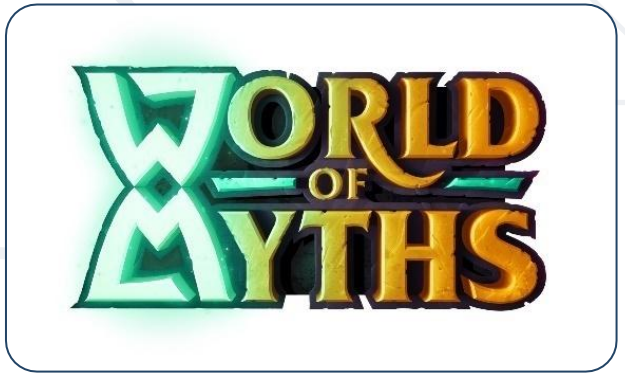
SoftUni Diamond Partners

Software University

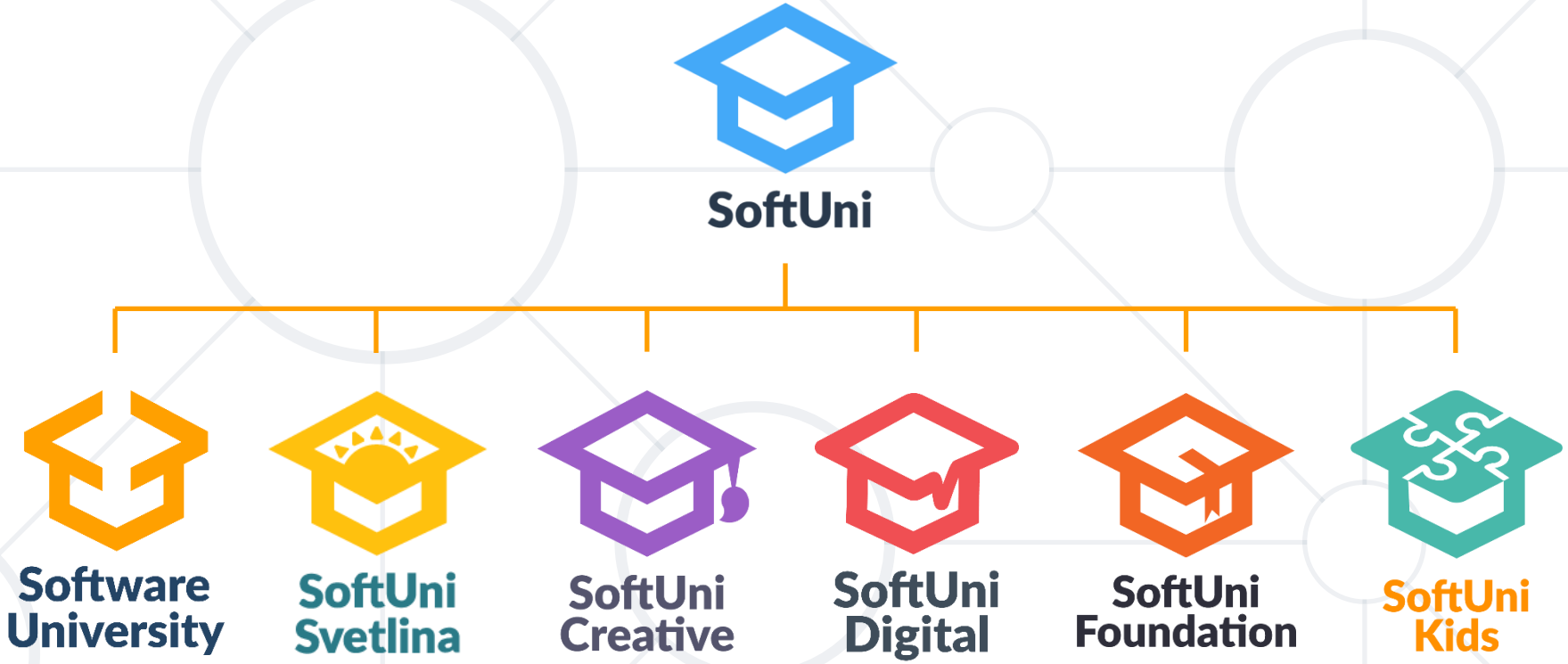


SoftUni Organizational Partners

Software University



Questions?



License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://softuni.org
- © Software University – https://softuni.bg

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, softuni.org
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg